

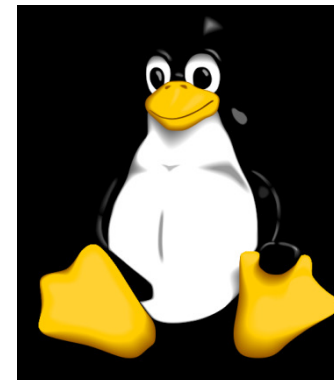
# Introduction into Linux, servers and other basics

Simon Domke

Practical course: The bioinformatics lab (SoSe 2012)

# What is Linux?

- free/open source multi-user operating systems
- based on the Linux kernel (developed and maintained by Linus Torvalds)
- UNIX like (Linux = **Linus** + **UNIX**), but no real derivative like BSD, Solaris or MacOS



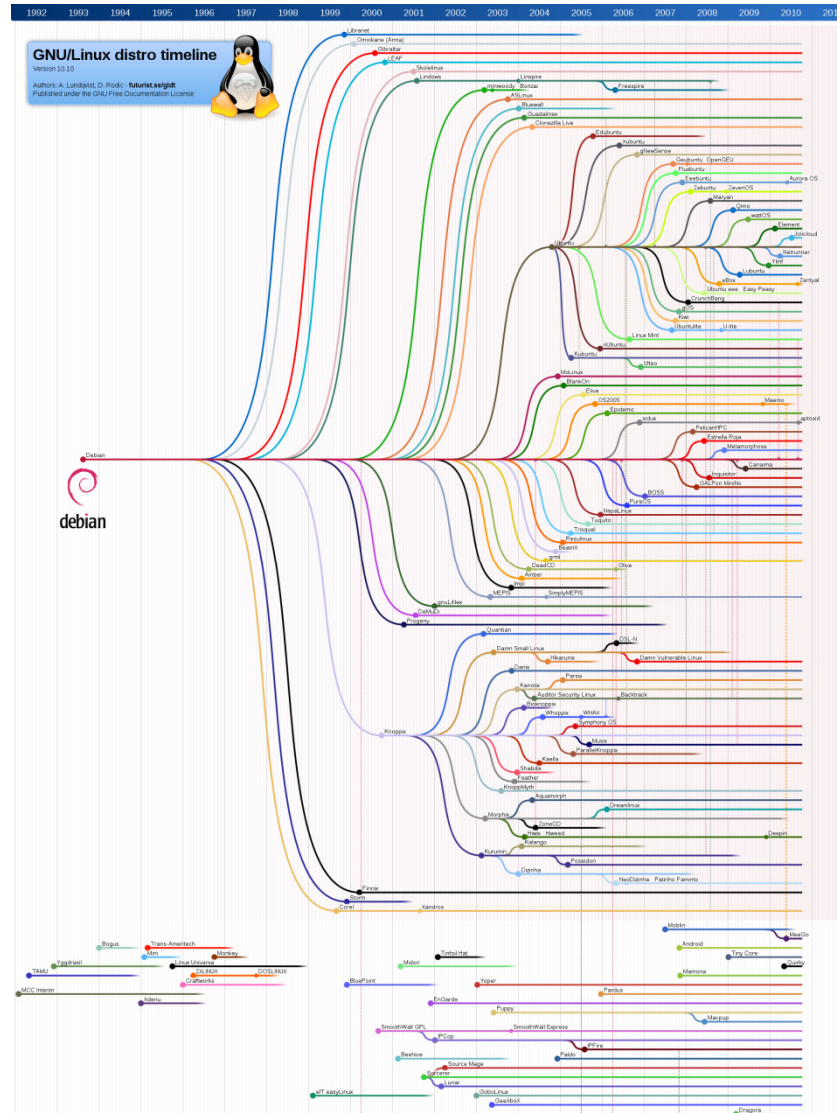
# What is Linux? – kernel development

- kernel.org: latest stable version 3.3
- developers commit changes to kernel in a phase called *merge window*
- changes are reviewed by maintainers (Linus Torvalds and others responsible for certain branches)
- new kernel version is released after changes are accepted

# What is Linux? - technique

- Linux kernel is monolithic (cmp. microkernel)
  - drivers and other modules are working directly in kernel space
  - faster but maybe also more error-prone
- provides API access to hardware
- schedules process execution and memory management

# Who uses the Linux kernel?



# What are the leading linux distributions?

- <http://distrowatch.com/>:

| Rank | Name                      | Page hits per day |
|------|---------------------------|-------------------|
| 1    | <a href="#">Mint</a>      | 4339              |
| 2    | <a href="#">Ubuntu</a>    | 2061              |
| 3    | <a href="#">Fedora</a>    | 1743              |
| 4    | <a href="#">openSUSE</a>  | 1613              |
| 5    | <a href="#">Debian</a>    | 1426              |
| 6    | <a href="#">Arch</a>      | 1186              |
| 7    | <a href="#">Mageia</a>    | 1157              |
| 8    | <a href="#">CentOS</a>    | 1074              |
| 9    | <a href="#">Puppy</a>     | 963               |
| 10   | <a href="#">PCLinuxOS</a> | 806               |
| 11   | <a href="#">Ultimate</a>  | 737               |
| 12   | <a href="#">FreeBSD</a>   | 735               |
| 13   | <a href="#">Lubuntu</a>   | 721               |
| 14   | <a href="#">Sabayon</a>   | 672               |

# What is free software?

- Free software ...
  - can be used, studied, and modified without restriction
  - can be copied and redistributed in modified or unmodified form either without restriction, or with restrictions
  - further recipients have the same rights under which it was obtained

# What is free software?

- free software is not completely free (ie. public domain)
- constraints:
  - derived software must use same license
  - license notice and author must be supplied
- free software can be sold (ie. does not have to be *free* of charge)
- Licenses: (L)GPL, Apache, BSD style, Artistic, ...



# GNU – General Public License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.



# What does this mean for me?

- source code of free software must be available
  - > you can hack it and adapt it to your needs
- recent problem on our servers:
  - Windows and Linux clients
  - LDAP user database
  - Samba mimicking Windows DC (accessing LDAP)
  - password fields separate in OpenLDAP:
    - „userPassword“: SHA hash of password
    - „sambaNTPassword“: MD4 hash of UTF-16LE password
- how to change password for both clients???



# What does this mean for me?

- Samba => Unix: just setting one line in Samba config, yeah!
- Unix => Samba (using passwd(1)): passwd uses PAM (man pam), doesn't know about databases
  - > adding pam\_ldap.so to stack: unix pw changed
  - > adding pam\_smbpass.so to stack: nothing happens, only local Samba database changed
- solution: just add the function myself!

```

93a94,97
> #define _GNU_SOURCE
> #include <asm/types.h>
> #include <openssl/md4.h>
>
333a338,437
> /* XXX: Samba pw change stuff */
> #ifndef false
> #define false 0
> #endif
> #ifndef true
> #define true 1
> #endif
>
> /* following came from the other byteorder.h to avoid include conflicts */
> #define CVAL(buf,pos) (((unsigned char *) (buf))[pos])
> #define SSVALX(buf,pos,val) (CVAL(buf,pos)=(val)&0xFF,CVAL(buf,pos+1)=(val)>>8)
> #define SSVAL(buf,pos,val) SSVALX((buf),(pos),((__u16)(val)))
>
> static int mdfour(unsigned char *md4_hash, unsigned char *link_str, int link_len);
> int mdfour(unsigned char *md4_hash, unsigned char *link_str, int link_len)
> {
>     md4_hash = MD4(link_str, link_len, NULL);
>     return 1;
> }
>
> static int _my_wcslen(__u16 *str);
> int _my_wcslen(__u16 *str)
> {
>     int len = 0;
>     while (*str++ != 0)
>         len++;
>     return len;
> }
>

```

.....

# What are the Debian releases?

- stable: officially released, stable distribution for production environment (current codename: *squeeze*)
- testing: new packages in testing phase are available in this release (currently: *wheezy*)
- unstable: development release (currently: *sid*)

# What is in a computer in terms of hardware?

- CPU
- RAM
- read-only storage
- HDD (rw storage)
- mainboard
- PSU
- extension cards

# How does a server differ in terms of hardware?

- more powerful and reliable hardware
  - higher internal CPU caches
  - higher number of physical cores
  - ECC RAM (error-correcting code)
  - redundancy: power supplies, hard disks, NICs (keywords: RAID, bonding)
  - different cases: often 19" rack mount

Where can you boot a (server) computer from for installing an operating system?

- read-only memory (CD, DVD, BD)
- removable hard drives (flash disks)
- network (TFTP in combination with DHCP/BOOTP and PXE compatible NICs)
- files (as we did on our VMs)

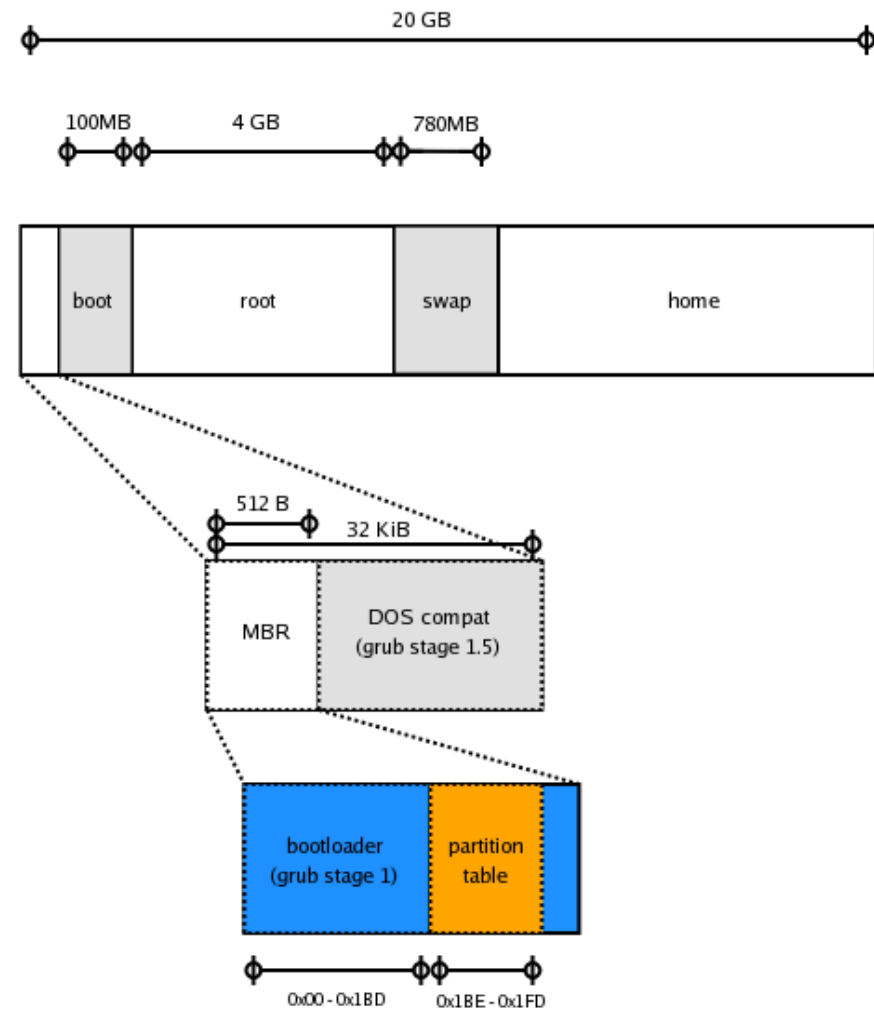


# What happens when you install an operating system?

- setup asks for a lot of configuration details
- disk is partitioned (ie. logically divided into parts) and formatted with a filesystem
- OS files are copied (kernel, GUI, shell, tools, applications, games, etc...)
- but how does the computer get to load and execute these files?

# The boot loader

- *Master Boot Record*
  - 512B total size
  - 446B boot loader
  - 64B partition table
  - 2B signature (0xAA55)
- after MBR: DOS compatibility region



# Boot process

- BIOS verifies MBR on disk (check signature)
- statically loads stage 1 boot code in MBR
- boot loader jumps to next stage (1.5, usually in DOS compat. region)
- open filesystem at specified partition and execute stage 2 code
- load ramdisk and tell kernel about it

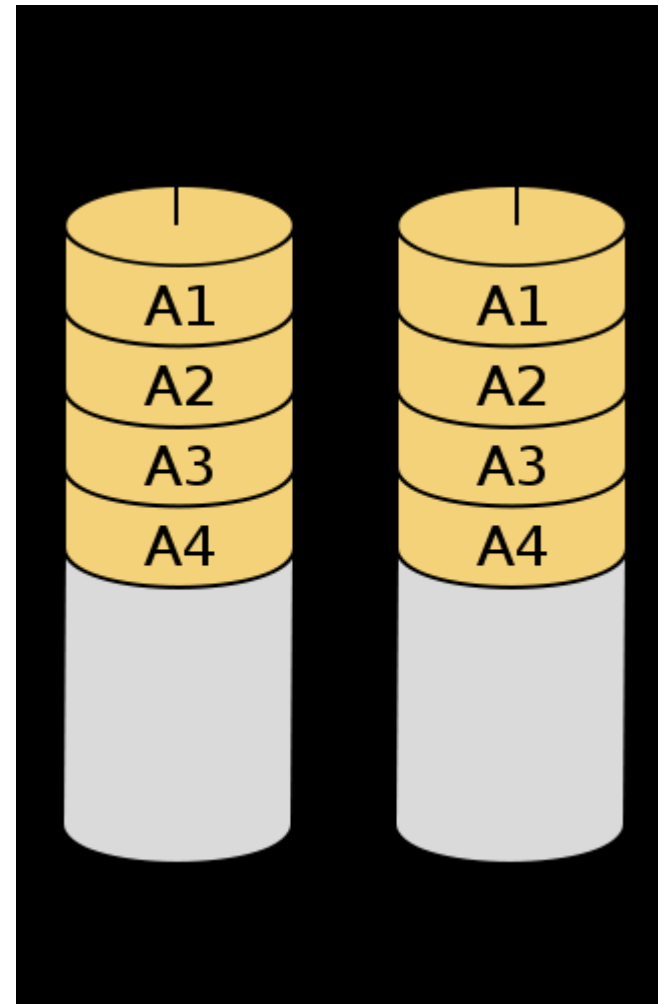
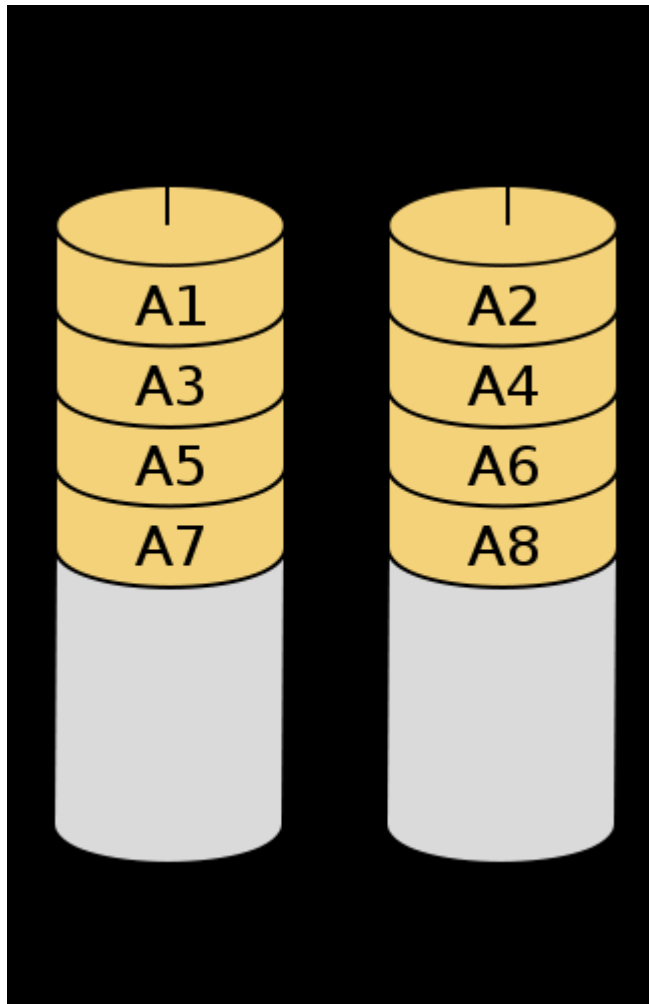
# How to distribute 24 disks for 6 persons working on 6 projects?

- combine multiple physical disks to one logical
  - *Redundant Array of Independent Disks* (RAID)
  - *Just a Bunch Of Disks* (JBOD)
  - hardware or software based (PU on extension card or CPU)
- RAID uses some space for parity, ie. unusable but enables recovery on disk failure

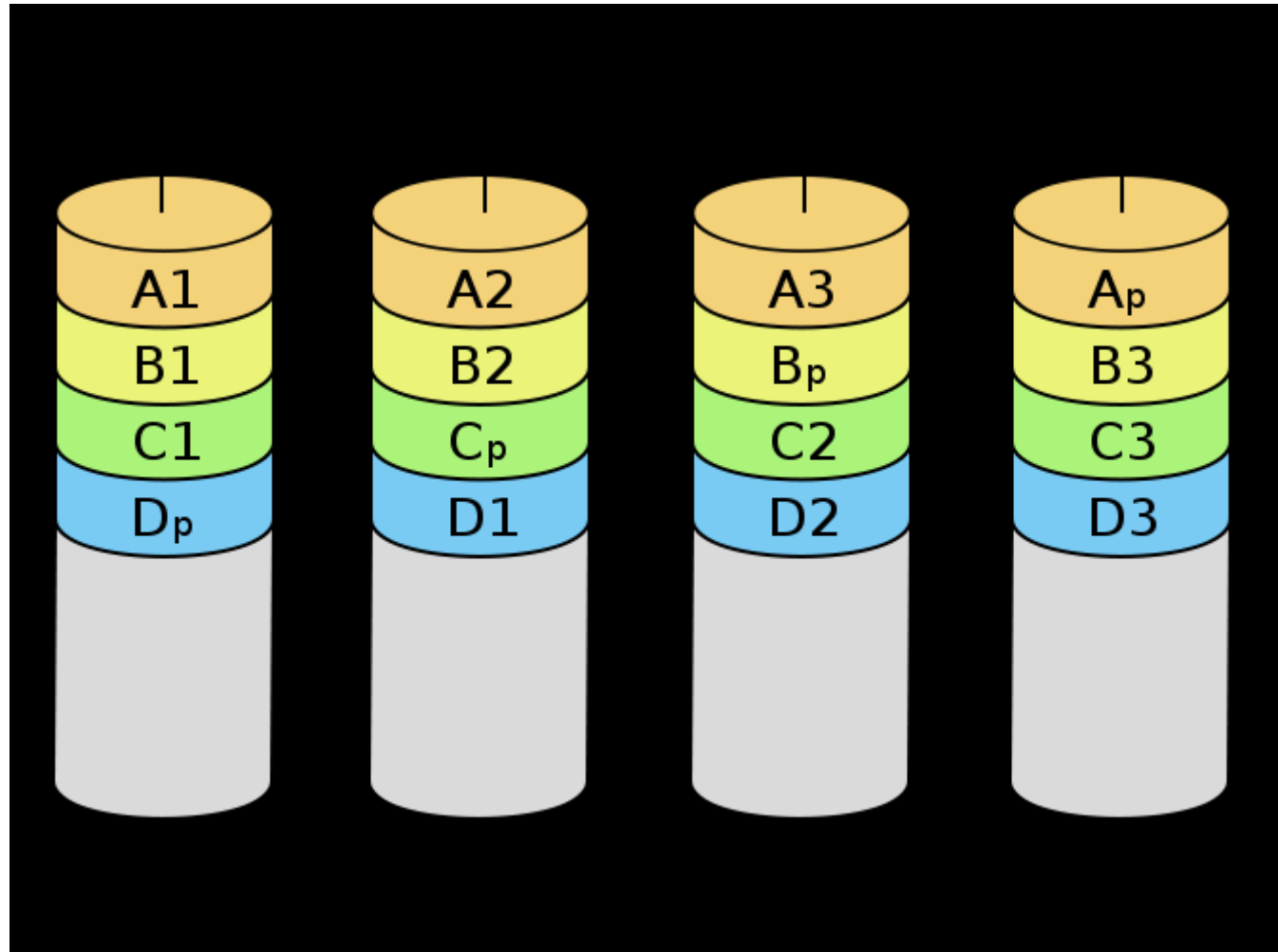
# RAID levels

- RAID 0 (striping): distribute data on multiple disks (+speed, ---fault tolerance)
- RAID 1 (mirroring): write data to multiple disks (~speed, +\* fault tolerance)
- RAID 5 (striping + parity):  $\# \text{disks} - 1 * (\text{size of smallest disk})$  usable space
- RAID 10/: RAID 0 (striping) of 2 RAID 1 (mirroring)
- many more

# RAID levels 0 and 1



# RAID level 5



# Make it available to others

- network file systems: e.g. NFS
- mountable from allowed hosts like any partition
  - /etc/fstab: `servername:/exported/path nfs ....`
- userrights:
  - users on client have to be known on server
    - system users
    - LDAP with name caching daemon (like nscd)



# NFS exports - example

```
/storage/home
10.150.255.0/27(rw,async,no_subtree_check,no_root_squash)
10.150.255.64/26(rw,async,no_subtree_check)
/storage/mirror/acng
install.olydorf.mhn.de(rw,async,no_subtree_check,no_root_squash)
/storage/mirror/images
install.olydorf.mhn.de(ro,async,no_subtree_check,no_root_squash)
/storage/apps
10.150.255.0/27(rw,async,no_subtree_check,no_root_squash)
10.150.255.64/26(ro,async,no_subtree_check)
/storage/group
10.150.255.0/27(rw,async,no_subtree_check,no_root_squash)
10.150.255.64/26(rw,async,no_subtree_check)
/storage/music
10.150.255.0/27(rw,async,no_subtree_check,no_root_squash)
10.150.255.64/26(ro,async,no_subtree_check)
/storage/projects/software
web.olydorf.mhn.de(rw,sync,no_subtree_check,no_root_squash)
/storage/backup
10.150.255.0/27(rw,async,no_subtree_check,no_root_squash)
10.150.255.64/26(ro,async,no_subtree_check)
```



# LVM – Logical Volume Manager

- logical partitions grouped in so called *volume groups*
- partitions resizeable
- hot swapping (share contents across disks)
- snapshots (backups)

# LVM - example

--- Logical volume ---

|                    |  |
|--------------------|--|
| LV Name            | /dev/lvm-raid/lvm-home                 |
| VG Name            | lvm-raid                               |
| LV UUID            | 2hJbzL-HRWK-anog-J9x2-GL61-6fqt-S5wRRi |
| LV Write Access    | read/write                             |
| LV Status          | available                              |
| # open             | 1                                      |
| LV Size            | 1,22 TiB                               |
| Current LE         | 10000                                  |
| Segments           | 1                                      |
| Allocation         | inherit                                |
| Read ahead sectors | auto                                   |
| - currently set to | 6144                                   |
| Block device       | 253:0                                  |